06-09-06

AFC
ZTW

TRANSMITTAL OF APPEAL BRIEF

Docket No.
03343/100I046-US1

In re Application of: Cuie Zhao

Application No.	Filing Date	Examiner	Group Art Unit
10/017,495-Conf. #5644	December 14, 2001	L. Truong	2194

Invention: NAME SERVICE OBJECT CLUSTERING

TO THE COMMISSIONER OF PATENTS:

Transmitted herewith is the Appeal Brief in this application, with respect to the Notice of Appeal filed: March 7, 2006 .

The fee for filing this Appeal Brief is $ 500.00 .

[X] Large Entity [ ] Small Entity

[ ] A petition for extension of time is also enclosed.

The fee for the extension of time is _____ .

[X] A check in the amount of $ 500.00 is enclosed.

[ ] Charge the amount of the fee to Deposit Account No. 04-0100 .

[ ] Payment by credit card. Form PTO-2038 is attached.

[X] The Director is hereby authorized to charge any additional fees that may be required or credit any overpayment to Deposit Account No. 04-0100 .

Dated: June 7, 2006

Melvin C. Garner
Attorney Reg. No. : 26,272
DARBY & DARBY P.C.
P.O. Box 5257
New York, New York 10150-5257
(212) 527-7717

Express Mail Label No. Dated: _____

Docket No.: 03343/100I046-US1
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Cuie Zhao et al.

Application No.: 10/017,495 Confirmation No.: 5644

Filed: December 14, 2001 Art Unit: 1226

For: NAME SERVICE OBJECT CLUSTERING Examiner: Truong, Lechi

APPELLANTS' BRIEF ON APPEAL UNDER 37 C.F.R. § 41.37

MS Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Appellants submit this brief in accordance with 37 C.F.R. § 41.37 in support of their appeal from the final Office Action mailed September 7, 2005 by Examiner Lechi Truong in the above-identified patent application.

On March 7, 2006, a Notice of Appeal was filed in this case. Concurrently, a Pre-Appeal Brief Request for Review was filed. A Notice of Panel Decision from Pre-Appeal Brief Review was mailed on May 8, 2006, resetting the time period for filing an Appeal Brief to one month from mailing of the decision. In accordance with 37 C.F.R. § 41.37, this brief is filed within one month of the May 8, 2006, Notice of Panel Decision from Pre-Appeal Brief Review, and is in furtherance of said Notice of Appeal.

I. REAL PARTY IN INTEREST

The real party in interest for this appeal is Borland Software Corporation. Rights in and to this application have been assigned to Borland.

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

There are 24 claims pending in application, i.e. claims 1-24. They are reproduced in the **Claims Appendix**. The current status of the claims is as follows:

1. Claims canceled: none;

2. Claims withdrawn from consideration, but not canceled: none;

3. Claims pending: 1-24;

4. Claims allowed: 3-8 and 11;

5. Claims rejected: 1, 2, 9, 10, and 12-24.

Claims 1, 14, 23, and 24 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of U.S. Patent 6,209,018 to Ben-Shachar et al. ("Ben-Shachar"), in view of U.S. Patent No. 6,606,643 to Emens et al. ("Emens").

Claims 2 and 9 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of Ben-Shachar, Emens, and the publication entitled Java Reflective Broker ("JR").

Claims 10, 12, and 13 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of Ben-Shachar, Emens, and the Arno webpage "WG:[omiORB] load balancing example" ("Arno").

Claims 15 and 16 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of Ben-Shachar, Emens, U.S. Patent No. 5,675,795 to Rawson et al. ("Rawson"), and U.S. Patent No. 5,452,445 to Nelson et al. ("Nelson").

Claim 17 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of Ben-Shachar, Emens, Rawson, Nelson, and JR.

Claims 18 and 19 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of Ben-Shachar, Emens, Rawson, Nelson, and Arno.

Claims 20-22 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of Ben-Shachar, Emens, and Rawson, Nelson, and U.S. Patent No. 5,742,759 to Nesset et al. ("Nesset").

The claims on appeal are claims 1, 2, 9, 10, and 12-24.

For the purpose of the present appeal, Appellants request that the claims be considered to form a single group.

IV. STATUS OF AMENDMENTS

In response to the final Office Action mailed September 7, 2005, Appellants filed a Request for Reconsideration on November 7, 2005. The Request for Consideration traversed the rejection by presenting arguments distinguishing the pending claims over the cited art, without making any amendments. The Examiner responded thereto in an Advisory Action mailed December 13, 2005. In the Advisory Action, the Examiner indicated that the Request for Reconsideration did not place the application in condition for allowance and did not enter the submission.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present invention is directed to a method of transparent fault tolerance, failover, and load balancing of Common Object Request Broker Architecture (CORBA) object servers. Name service clusters are established for the object servers, each having unique object binding table containing object server references. (Specification, page 7, lines 2-4.) By clustering object references, the naming service is able to load balance and provide fault tolerance and failover for the set of object references associated with the clustered name. (Specification, page 5, lines 18-20.) Thus, there is no need to recreate name service clusters with each call to a name, since the load balancing is performed among a common predetermined group of object reference binders. (Specification, page 5, lines 20-21.)

When a client invokes a name service cluster, a load balance is performed by the name service by selecting an object reference located in the invoked name service cluster. (Specification, page 7, lines 7-10.) The name service cluster then returns the object reference previously bound to the cluster so that the client may communicate with the server associated with the object reference selected. (Specification, page 7, lines 7-10.) The name service is inherent to the CORBA infrastructure, thus, load balancing occurs transparently to the user, i.e., without requiring any knowledge or program design on the part of the user.

If a server becomes unresponsive or performs inadequately, the present invention also provides for transparent fault tolerance and failover to another server. When a client invokes a name service cluster, a cluster component is created and added to the resolved object reference such that if a failure occurs, the information in the cluster component will permit the client ORB to contact the cluster object and obtain another object in the same cluster to which the client ORB can then failover. (Specification, page 8, lines 14-18.) By providing an object reference to the cluster, if an attempt is made to send a request to a failed server through the stale object reference, the client ORB associated with the stale object reference will transparently intercept the cluster component and transparently select another object reference in the same cluster. (Specification, page 8, lines 21-26.) The new object reference is returned to the client, thereby establishing communication

between the client and the server through the newly returned object. (Specification, page 8, lines 21-26.)

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether the Examiner erred in holding that claims 1, 2, 9, 10 and 12-24 are unpatentable under 35 U.S.C. §103(a) as set forth above.

VII. ARGUMENT

Independent claims 1 and 15 are patentable over the cited combinations, because Ben-Shachar, alone or in combination, neither discloses nor suggests the notion of user transparency with respect to load balancing, fault tolerance, and failover.

The claimed invention is a "computer implemented method of fault tolerance, load balance and failover of CORBA object servers . . . wherein the fault tolerance, the load balance and the failover are preformed transparently," as recited by independent claims 1 and 15. The present invention distinguishes over the cited prior art because none of the references, alone or in combination, disclose or suggest a transparent method of fault tolerance, load balance, or failover of CORBA object servers.

With the present invention, the client code, or application code, interfacing a system implementing the method of the claim invention does not need to be aware of the fault tolerance, failover or load balancing system. The transparency is simply "accomplished using a flag located in a file, such as a configuration file [of the naming service], or the like." (Specification, page 5, lines 10-11.) By establishing name service clusters for object services and clustering object references within the sane name service cluster, the naming service load balances and provides fault tolerance and failover for the set of object references associate with the clustered name. (Specification, page 5, lines 18-20; page 7, lines 2-4.) In the present invention, merely invoking the clustered naming

service provides a software client the transparent load balancing, fault tolerance and failover of the present invention. These features are transparent to the software and the application developer.

The Examiner's rejection of the claims on appeal incorrectly relies on Ben-Shachar as disclosing transparent fault tolerance, load balancing, and failover corresponding to the claimed invention. (Final Office Action, Sept. 7, 2005, *citing* Ben-Shachar, column 10, lines 3-7 and column 11, lines 60-63.) The Examiner explains the bases of rejection in the Advisory Action, mailed December 13, 2005, noting that

> Ben-Shachar teaches the procesing [sic] of obtaining a worker is transparent to the client, because the service proxy performs the necessary steps to obtain the worker (col 1, ln 5-7), the proxy service cand [sic] bind to the service located 84 (col 7, ln 35-38). . . . Moreover, this entire process is transparent to the client because the client's proxy is simply returns a service handle (col 20, ln 14-16.)

(Advisory Action, Page 3, Para. 2.)

The Examiner misinterprets the transparency provided by Ben-Shachar. The transparency provided by Ben-Shachar is entirely within the context of the additional service proxy, service locator, and service manager required by Ben-Shachar. (Ben-Shachar, column 10, lines 5-8.) The transparency of Ben-Shachar relates only to obtaining a worker. Thus, the transparency of Ben-Shachar is completely unrelated to load balancing, fault tolerance, and failover, as expressly recited in claims 1 and 15. Moreover, achieving any form of transparency in Ben-Shachar's framework requires the implementation of a service proxy, service locator, and service manager, and all service calls must be routed through these components. (Ben-Shachar, column 7, lines 25-30.) The client must invoke the service proxy, not the usual CORBA object, to obtain a worker to perform distributed communication. Thus, because Ben-Shachar requires the application developer to create new objects and new communication patterns, the system of Ben-Shachar is not truly transparent to either the developer or the application.

Systems based on Ben-Shachar's framework cannot simply plug in a CORBA implementation and obtain transparent load balancing, fault tolerance, and failover. Rather a system developer and designer must build the system to make it load balance and fault tolerance aware by

using service proxies, service locators, and service managers. Ben-Shachar clearly states that the disclosed system is a "service framework that provides significant extension to CORBA-based distributed object network system." (Ben-Shachar, column 9, lines 9-12.) Ben-Shachar requires infrastructure and services above and beyond the standard CORBA infrastructure. Thus, unlike the claimed invention, Ben-Shachar must be built on top of, and around, a CORBA based distributed system.

In contrast, the present invention, expressly and purposely, constrains itself to the existing CORBA communication pattern and does not require the introduction of new communication models, structure, or methods. Transparent load balancing, fault tolerance, and failover are provided through a CORBA Name Service, which is a standard component of a CORBA infrastructure. Thus, a distributed computer system that utilizes a CORBA Name Service, but does not provide load balancing, fault tolerance, and failover, can gain these features transparently, merely by plugging in a Name Service that implements the claimed invention. The claimed invention does not require the inclusion of a service proxy, service locator, or service manager, nor does it require any changes to the communication patterns between objects in the distributed computer system.

In contrast, Ben-Shachar's system requires participants to significantly change their existing communication patterns to enable load balancing. Rather than merely instructing a client to talk to a name service, as is done in a conventional CORBA implementation and the present invention, a client using the Ben-Shachar system must talk through a service proxy to a service locator, create a service, allocate a worker and then obtain the server. Ben-Shachar eliminates the standard CORBA Name Service and replaces it with a self-created Service Locator. The fault tolerance and load balancing must be added to the service proxy and thus is not transparent to the application or the developer. (Ben-Shachar, column 1, lines 49-63.) The present invention does not require the client or server to change their existing communication patterns because the transparent load balance and failover is provided within the CORBA name service.

Ben-Shachar not only fails to disclose the notion of transparency, in the sense that nothing needs to be added on top of the standard CORBA structure, but also specifically teaches away from it. Ben-Shachar's requirement to build or change a system to make it load balancing or fault tolerance aware clearly classifies Ben-Shachar as intrusive rather than transparent. Ben-Shachar's requirement of the service proxy, service locator, service object, and worker all cooperating with each other as part of a close knit framework built on top of CORBA make it impossible to perform the load balance and fault tolerance where a CORBA client and server both are unaware of the framework. Since the entire system must be built on the framework, Ben-Shachar established load balancing and fault tolerance cannot possibly be performed transparently.

Dependent claims 2, 9, 10, 12-4 and 16-25 are patent over Examiner's cited combinations.

The Examiner rejected claim 2, 9, 10, 12-14, and 16-24 over Ben-Shachar in view of various combinations of Ben-Shachar, Emens, JR, Arno, Rawson, Nelson and Nesset.

With respect to claims 2, 9, 10, 12-14, and 16-24, Appellant submits that these claims depend directly or indirectly from either independent claim 1 or 15. None of the secondary references cited by the Examiner disclose or suggest the features demonstrated above to be missing from Ben-Shachar. Thus, Appellant submits that these claims should be allowed for at least the reasons discussed for the respective base claims and in view of their own further recitations.

In view of the foregoing, Appellant respectfully submits that claims 1, 2, 9, 10, and 12-24 would not have been obvious in view of the cited references within the meaning of § 103. Therefore, it is respectfully submitted that based upon the preceding discussion, the Examiner should be reversed.

For all of the reasons set forth above, the rejections of claim 1, 2, 9, 10, and 12-24 should be reversed. Appellants respectfully request that the application be remanded to the Primary Examiner with an instruction to withdraw the 35 U.S.C. § 103(a) rejection, and pass the case to allowance.

Please charge any fee, except for the Issue Fee, that may be necessary for the continued pendency of this application to our Deposit Account No. 04-0100.

Dated: June 7, 2006

Respectfully submitted,

By _____

Melvin C. Garner
Registration No.: 26,272
DARBY & DARBY P.C.
P.O. Box 5257
New York, New York 10150-5257
(212) 527-7700
(212) 527-7701 (Fax)
Attorneys/Agents For Appellants

APPENDIXES

CLAIMS APPENDIX

The following is a copy of the claims involved in the appeal:

1. (Previously Presented) A computer implemented method for fault tolerance, load balance and failover of CORBA object servers, comprising the steps of:

establishing name service clusters for the object servers which each contain a unique object binding table that contains object server references;

in response to a request from a client that invokes a cluster, performing a load balance by having the name service select an object server located in the invoked cluster;

appending a cluster component to the invoked cluster to provide failover upon failure of the object server;

forwarding a selected object reference to a client upon completion of the load balance; and

permitting the client to communicate with the server associated with the selected object reference which was forwarded to the client, wherein the fault tolerance, the load balance and the failover are performed transparently.

2. (Original) The method of claim 1, said invoking step comprising the step of:

binding to the server using an IP Address and port number contained in the specific object reference.

9. (Previously Presented) The method of claim 1, further comprising the step of:

specifying a load balance algorithm upon establishing the naming service cluster to perform name service load balancing of object references contained within the clusters.

10. (Original) The method of claim 1, wherein said load balancing is performed based on a predetermined method.

12. (Original) The method of claim 1, wherein said load balancing is performed based on a predetermined method.

13. (Original) The method of claim 12, wherein the predetermined method is a Round robin load balancing algorithm.

14. (Previously Presented) The method of claim 1, wherein the object binding table of each cluster contains object references;

wherein each object server reference represents a single server.

15. (Previously Presented) A computer implemented method for fault tolerance, load balance and failover of CORBA object servers, comprising the steps of:

establishing name service clusters for the object servers which each contain a unique object binding table that contains object server references;

setting a flag in a file to activate the implicit clustering;

in response to invoking a cluster contained in a context having clusters; performing a load balance by having the name service select an object server located in the clusters;

forwarding a selected object reference to a client upon completion of the load balance; and

communicating with the server associated with the selected object reference which was forwarded to the client, wherein the fault tolerance, the load balance and the failover are performed transparently.

16. (Original) The method of claim 15, wherein the file is a configuration file.

17. (Original) The method of claim 15, said invoking step comprising:

binding to the server using an IP Address and port number contained in the specific object reference.

18. (Original) The method of claim 15, wherein said load balancing is performed based on a predetermined method.

19. (Original) The method of claim 18, wherein the predetermined method is a Smart Round Robin load balancing algorithm.

20. (Original) The method of claim 15, wherein object reference binding having identical names are clustered together in common clusters such that a common group of object reference binders servers is created.

21. (Original) The method of claim 20, further comprising the step of:

specifying a load balance algorithm to perform load balancing of object references contained within the common group of group of object reference binders.

22. (Original) The method of claim 21, wherein initially the load balance algorithm is Smart Round Robin.

23. (Previously Presented) A computer implemented method for transparently load balancing CORBA object servers, comprising the steps of:

establishing name service clusters for the object servers which each contain a unique object binding table that contains object server references;

in response to a request from a client that invokes a cluster, the name service for the cluster performs a load balance to select an object server located in the invoked cluster which can handle the request;

forwarding a selected object reference to a client upon completion of the load balance; and

permitting the client to communicating with the server associated with the selected object reference that was forwarded to the client.

24. (Previously Presented) The method of claim 23 further including the step of appending a cluster component to the invoked cluster, such cluster providing information to the client as to a failover server to be accessed upon failure of the object server.

EVIDENCE APPENDIX

All evidence is in the record.

RELATED PROCEEDINGS APPENDIX

There are no related proceedings for this matter.

Application No. (if known): 10/017,495 - Conf. #5644 Attorney Docket No.: 03343/100I046-US1

Certificate of Express Mailing Under 37 CFR 1.10

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail, Airbill No. EV835933044US in an envelope addressed to:

MS Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

on June 7, 2006
Date

L. Stantini
Signature

L. Stantini
Typed or printed name of person signing Certificate

Registration Number, if applicable

Telephone Number

Note: Each paper must have its own certificate of mailing, or this certificate must identify each submitted paper.

Appeal Brief Transmittal (1 page)
Appellants' Brief On Appeal Under 37 C.F.R. §41.37 (15 pages)
Check No. 12007 in the amount of $500.00
Return Receipt Postcard